DTS
DEPARTMENT OF TECHNOLOGY SERVICES

TECHNICAL ARCHITECTURE REVIEW

Project Name:	**Alfresco Document Management**
Requestor:	John Angus
Date of Initial Request:	December 2007
Request Description:	Is Alfresco a viable document management (DM) solution for departmental document management solutions, either as a component of an application such as e-Rules or as a standalone DM repository?
Agency or Agencies:	DAS, Public Services Commission, Enterprise
Reviewers:	Bob Woolley and Dave Fletcher
ARB Acceptance Date:	
Agency Requestor Acceptance Date:	

Introduction

The State has implemented a variety of document management solutions over the last decade. These solutions are largely proprietary and represent substantial costs to the State. Adoption of these solutions has been constrained by cost and implementation complexities, so only a small percentage of agency document management requirements have been addressed. Open source has become a viable option over the last several years. This review is limited to Alfresco, one of the leading open source vendors for Document Management (DM).

Objectives and Scope of Review

This review focuses primarily on the DM capabilities of the Alfresco product with minimal consideration of the emerging Web Content Management (WCM) and Records Management (RM) capabilities that have emerged since 2007.

Product Overview and Architecture

Alfresco was founded in 2005 by John Newton, co-founder of Documentum, and John Powell, former COO of Business Objects. Investors include the investment firms Accel Partners and the Mayfield Fund. The original technical staff consisted of engineers from Documentum and Oracle. Initially focused on document management, in May of 2006, Alfresco announced its intention to expand into Web content management by acquiring senior technical and managerial staff

from Interwoven. In 2007, Alfresco hired the principal sales engineer from Vignette. Alfresco is now offering a comprehensive Electronic Content Management solution that includes Document Management (DM), Web Content Management (WCM), and Records Management (RM).

Alfresco was released as an open source product and has gained broad acceptance for document management in a variety of enterprises. The Alfresco architecture is illustrated in Figure 1.

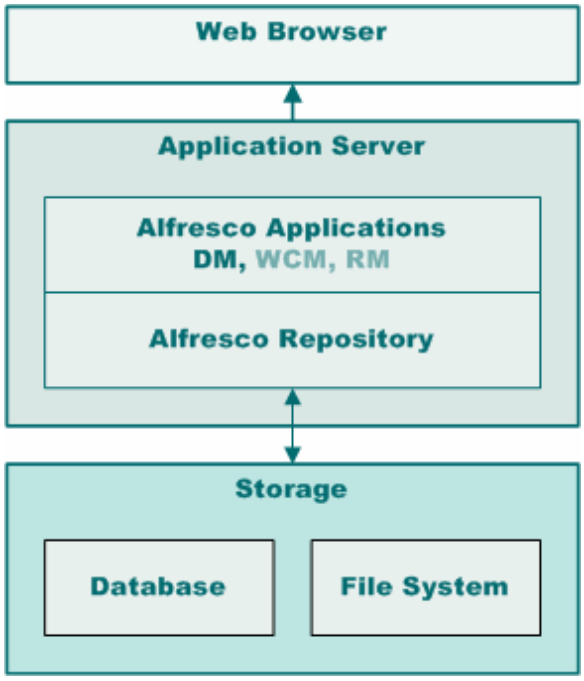

Figure 1. Alfresco Repository Architecture

Figure 1 is typical of a standard Web architecture, where an application server houses the logic for both the user interface and domain model. Storage of data and content is provided by persistent back-ends, such as a database or file system. Any number of Web browsers can connect to the application without installing anything on the client.

In Figure 1, the application server houses both the Alfresco Application and the Alfresco Repository. An Alfresco Application provides a complete solution tailored to a specific area of Content Management, such as Document Management (DM), Web Content Management (WCM), or Records Management (RM). The Alfresco Repository provides a set of reusable cross-cutting content management services, such as content storage, query, versioning, and content transformation. These services may be utilized by multiple applications.

Figure 2 illustrates the component architecture of the Alfresco repository. The Alfresco Repository is responsible for the storage and retrieval of content. This responsibility is provided by the following three Foundation Services:

- Nodes provide metadata and structure to content. A node may support properties (e.g., author) and relate to other nodes (e.g., represent folder hierarchies or annotations).

- Content is the actual information being recorded (e.g., a Word document, a scanned image, a PDF file, or XML fragment). Metadata and content may be structured according to the rules defined in a Content Model. For example, the Alfresco Document Management application relies on a model that describes folders and files.

- The Search Service handles indexing information and allows the retrieval of metadata and content via many different lookup options.

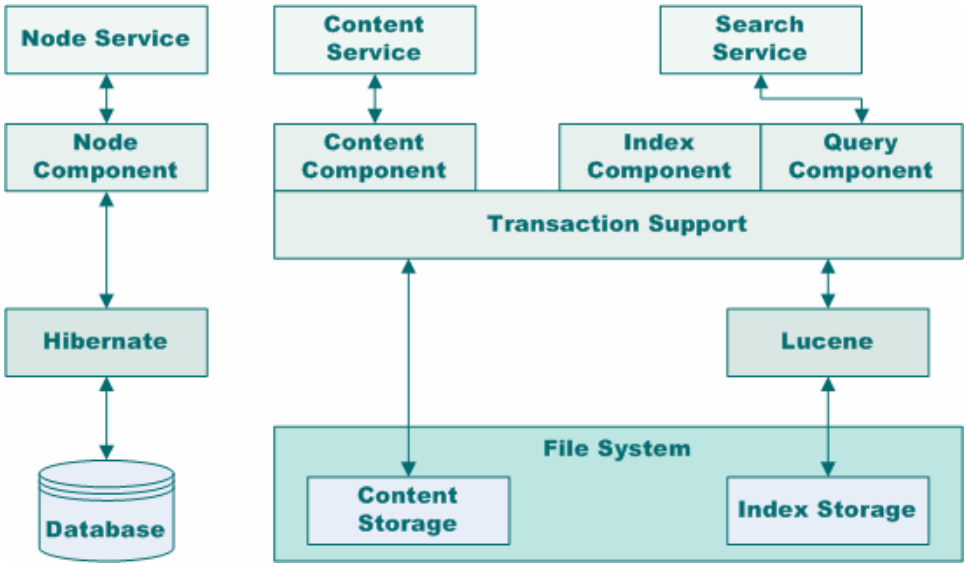

Figure 2. Alfresco Key Foundation Services

By default, Alfresco has chosen to store metadata in a database and content in a file system. Using a database immediately brings in the benefits of databases that have been developed over many years, such as transaction support, scaling, and administration capabilities. Content is stored in the file system to allow for very large content, random access, streaming, and options for different storage devices. This architecture permits scaling for very large content repositories[1] and

[1] *Benchmark Results: Scalability of Alfresco Content Management System in a Unasys ES700/one Enterprise Server Environment*, Unisys, December 2007.

associated metadata. A recent Unisys study[2] benchmarked DM performance in a repository of 107 million documents and reported excellent retrieval performance.

Building on the key storage and retrieval services, the Alfresco Repository provides:

- Content Transformation
- Metadata Extraction
- Templating
- Classification
- Versioning
- Locking
- Content Modeling
- Image Manipulation
- Workflow
- Import and Export Services
- Permissions

Protocols

Alfresco supports multiple protocols for access to the content managed by the repository including:

- CIFS (Common Internet File System);
- WebDAV; and,
- FTP.

All of these protocols expose the model of folders and files, which maps onto folder and file nodes held in the Repository. WebDAV and FTP are well known protocols, but CIFS deserves more attention. CIFS transforms the Alfresco Repository into a standard file system. Any tool that understands how to read and write to a file system also knows how to directly read and write to the Alfresco Repository. CIFS projects an actual file system enabling additional compatibility with the hosting operating system. For example, in Windows, it's possible to use Offline Synchronization and Briefcase features against the Alfresco Repository for offline work.

Application Programming Interfaces (APIs)

The Alfresco Repository actually provides three APIs. These API services are illustrated in Figure 3.

[2] *Ibid.*

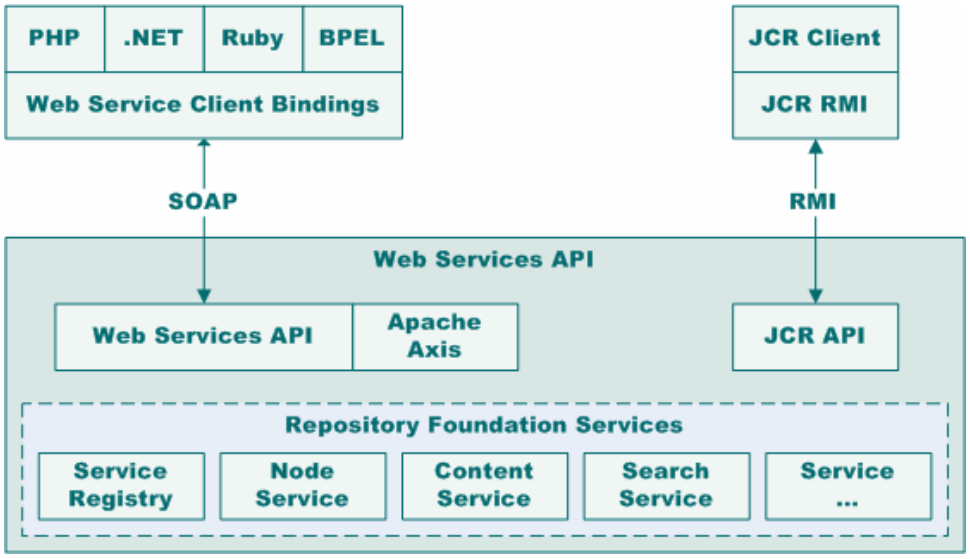

Figure 3. Alfresco API Services

Repository Foundation Services is a set of local Java interfaces covering all capabilities which are useful for clients who wish to embed the Repository into an application such as eRules. The other two APIs are built on top of the Foundation Services, and take advantage of content model logic and rules. The two other APIs are:

- **Java Content Repository (JCR) API (JSR-170):** JCR (Content Repository API for Java Technologies) is a standard Java API (as defined by JSR-170) for accessing content repositories. Alfresco provides support for JCR level 1 and 2, giving standardized read and write access. Supporting this API provides the following benefits:

    - Low risk—The Alfresco Repository can be evaluated and developed against, but swapped out with, another JCR repository if it does not fit customer requirements.
    - Familiarity—Developers who know JCR, know Alfresco.
    - Tools—Tools, clients, and other third party JCR solutions are immediately available to the Alfresco community.

    Alfresco JCR is implemented as a light facade on top of the Repository Foundation Services. So, although a familiar API is provided, it sits upon a fully transactional, secure, and scalable repository which supports many deployment options. Alfresco has announced plans to move toward JSR-283, the next version of the JCR.

- **Web Services:** This is the final API provided by the Alfresco Repository. This API supports remote access and bindings to any client environment, not just Java. For example, the Alfresco community is already using PHP,

Ruby, and Microsoft .NET. Numerous standards and integration efforts are focused around Web services. SOA is recognized as a way to integrate disparate enterprise systems including content management. BPEL plays an important role in orchestrating all of these services, and Alfresco supports the SOA architecture pattern.

Deployment Options

Alfresco can be deployed in a variety of the common environments supported by the State, such as Tomcat 5.5.x. A typical deployment is on a Linux server in a LAMP environment. Alfresco requires access to a Java Virtual Machine 6.x (JVM); MySQL or other suitable database environment; JDBC driver for MySQL; a Web browser such as Firefox 2.x; and, an FTP client. Deployments[3] can take place using any of the following patterns:

- **Embedded Repository**: An Embedded Repository is contained directly within a host where the host communicates with the Repository in the same process via the Repository Foundation Services. Typical hosts include content-rich applications that require content-oriented storage, retrieval, and services. eRules is an example of an application that embeds Alfresco services.

- **Repository Server or Application Server**: A Repository Server is a stand-alone server capable of servicing requests over remote protocols and providing appropriate responses. A single server can support any number of different applications and clients. New applications may be added arbitrarily. Deployments can choose which of the protocols are supported. Web server features, such as transaction management and resource pooling, are injected into the Repository so it can take advantage of them. The Repository automatically benefits from any enhanced features provided by higher-end Web application servers. For example, the Repository can be embedded inside Apache Tomcat for the lightest weight deployment, but it may also be embedded inside other J2EE compliant application servers to take advantage of distributed transactions. A Web application can also become the host for an embedded Repository, and remote access is via the application using HTTP.

- **Clustered Repository Server**: A Clustered Repository Server supports large numbers of requests by employing multiple processes against a single Repository store. Each Embedded Repository is hosted in its own Web server and the collection as a whole (i.e., the cluster) acts as a single Repository.

[3] *Alfresco Developers Wiki* at http://wiki.alfresco.com.

- **Hot Backup:** In this deployment, one Repository Server is designated the Master and another completely separate Repository Server is designated as the Slave. The live application is hosted on the Master, and as it is used, synchronous and asynchronous replication updates are made to the Slave as the backup server. The backup remains in read-only mode. If for some reason the Master breaks down, access is swapped to the Slave to continue operation.

The Alfresco repository is an enterprise-scale repository designed to be distributed, federated, and scalable. The system is architected to use aspects and Aspect-Oriented Programming wherever possible. The capabilities of the Alfresco repository are those found in many of the established commercial enterprise content management repositories. These capabilities include:

- aspect-oriented, including the ability to add new aspects;
- hierarchical folder structures;
- document types with standard document behaviors;
- metadata, with extensible types and complex properties;
- classifications, allowing users to browse or search based upon global definitions;
- rule-driven processing of content to add or modify data or move content;
- automated content processing, including auto-classification, auto-indexing, and workflow handling;
- global dictionary, a namespace driven dictionary that allows hierarchies and standardization of definitions;
- open authentication, which is configuration-driven to use enterprise standards for authentication, such as LDAP (UMD and e-Directory) and Microsoft Active Directory™;
- full text indexing and retrieval, which uses the Lucene search engine and content transformation to search many different content types;
- team collaboration spaces;
- versioning;
- locking, a configurable aspect to add locking, if necessary;
- content-streaming; and,
- configuration control, a central point of configuration for all Alfresco features.

Baseline of Current Architecture

At present there is one implementation of Alfresco that has been formally identified within the context of an approved IT project. This is represented by the eRules project. The architecture components for this project, as listed in the eRules application, will utilize and/or integrate the following software applications:

Framework/Tool	Description	Use
Struts 2.x	Web Application Framework	Front-end Web application.
Alfresco 2.x	Content Management Software	Document management solution.
OpenOffice.org 2.x	Office Tool Software	Document publication and transformation plug-in for Alfresco.
Hibernate 3.x	ORM Framework	Database support for Alfresco.
MySQL 4.1.2x	Database	Data storage.

This is an embedded implementation of Alfresco and uses currently supported environments such as Struts 2.X. It may be preferable to use the Spring and Hibernate environments that are directly supported by Alfresco and bypass the Struts 2.X layer for standard non-embedded repository implementations, such as the one anticipated by the Public Services Commission. This will result in less infrastructure coding and a simpler implementation for most agency repositories.

Best Practices Review

One of the more useful approaches to assessing document management best practices is from a maturity model perspective. In most cases DM implementations seek to enable collaboration and access to documents. DM maturity seeks to characterize the level of understanding of the people who work with documents. The table that follows lists a maturity model[4] that goes well beyond sharing and access from a common repository.

- Level 0: Documents are produced in isolation.
- Level 1: Working together and sharing documents.
- Level 2: Have a secure repository.
- Level 3: Consistent organization of documents.
- Level 4: Documents are usable and relevant.
- Level 5: Collaborative working with documents.
- Level 6: Managing and automating business processes.
- Level 7: Optimizing worker productivity.
- Level 8: Integration across enterprise applications.

From an aggregate maturity perspective the State appears to be between Level 1 and 2, with a few program exceptions that reach Level 3. Historically, government has spent a great deal of time producing documents in some degree of isolation; however, there has been substantial ad hoc sharing of document drafts between participants. Detailed best practices are available for each of these maturity areas.

[4] Goodwin, Mark, *The Enterprise Document Management Maturity Model*, McLaren Software, May 2006.

Emerging Technologies and Trends

Open source DM solutions with enterprise capabilities such as Alfresco represent a clear trend in this industry. There are more than 1,700 DM solutions available for work groups through enterprise applications. New and revised products are moving away from proprietary environments and toward open source environments such as Linux, Tomcat, JBoss, MySQL, and LDAP, to name a few. There seems to be a pronounced move away from overly complex solutions toward solutions that are easier to work with and consume. Documentum and Alfresco are both examples of this trend on the commercial and open source sides. Users want to be able to share, retrieve, and manage documents with minimal user complexity.

Financial Analysis

Alfresco is available as an open source community supported edition at no cost. A supported enterprise license is available for $15,000 per year. Test environments are $5,000 per year. Administrative Rules has already purchased an enterprise license, which may be sharable with other agencies. Deployment costs are relatively small, since most average deployments could take place within a shared LAMP environment. Dedicated server environments are available as needed, as are varying degrees of clustering and hot standby. Development costs for the repositories do seem to vary dramatically from normal application development that utilizes an existing infrastructure. Overall costs are negligible compared to large proprietary licensed products.

Security Review and Analysis

Alfresco provides integration and implementation of security across all aspects of the architecture and integrates fairly seamlessly with existing authentication and control environments. Security functionalities include:

- users and user management;
- provision of personal information about users;
- user authentication;
- groups and group management;
- ownership of nodes within the repository;
- repository wide permissions;
- permissions at the node level;
- an extendable permission model; and,
- access control, to restrict calls to public services to suitable authenticated users.

There do not appear to be any evident security concerns as long as the application is deployed following established State security procedures and practices.

Operational and Infrastructure Analysis

Standard Alfresco implementations will generally be deployed in an existing LAMP shared environment or in a dedicated environment for larger implementations. Database and storage environment requirements are consistent with existing infrastructure. There are no obvious operational and infrastructure issues for Alfresco deployment.

Solution Delivery Impact and Analysis

Alfresco has the potential to become a standard document management tool for agency and enterprise DM implementations. It can be embedded within applications and portals or implemented on a standalone basis as needed. It would be advantageous for development staff to become proficient with Spring and Hibernate, since both are leading environments for application development nationally.

Agency Services Impact and Analysis

Alfresco does not have any apparent impact on agency services with the exception that some agencies will need to migrate away from the soon to be discontinued GroupWise DM solution to another DM environment. Tools are available to migrate GroupWise document repositories and metadata to Alfresco.

Summary and Recommendations

Alfresco provides a feature rich DM environment at a relatively low cost of entry for interested agencies. The application is advantageous in that it can be implemented as a repository with agency specific Web interfaces or as an embedded product within a specific application that needs a DM solution. Reported scalability and reliability data indicate that the application can handle large document libraries. Alfresco uses stable broadly accepted open source technology components such, as Spring and Hibernate. The application can be implemented within php, .net, Ruby, Struts, and Java. Web services and JCR are both supported, as is connectivity, to a wide range of data sources. Deployment is possible as a standalone or under Tomcat and a wide range of application server environments currently used by the State. Reviews have praised the application for staying current with new technologies and deployment architectures. Overall, the application provides a stable and reliable DM environment that will meet many of the State's DM requirements. It is recommended that Alfresco be:

- approved as a standard alternative environment for agency and work group DM implementation;
- established as a hosted Alfresco DM Repository for agency document hosting using existing rates for hosting and storage;
- approved as a standard DM environment for embedded DM implementation within applications and portal environments; and,

- established as the “migrate to” environment for agencies that are using existing GroupWise DM functionality.

In addition, it is recommended that:

- DET development staff be provided additional training for Spring and Hibernate if this is consistent with future framework direction;
- The CTO office utilize Alfresco for the document repository for all Technical Architecture (TA) documents;
- other Alfresco functionality for WCM and RM be investigated for possible application for these functional areas within State agencies; and,
- that an engineering validation of these recommendations be completed by DET development staff.

The overall savings to agencies that can be accomplished by using Alfresco in a standard hosting and application server environment, such as LAMP, are significant and represent an opportunity for future cost avoidance. DM within the State is at a relatively low maturity level. A consistent DM approach using the Alfresco platform provides an opportunity to improve business processes for DM and raise the overall quality of handling and finding document information.

References

Alfresco Developers Wiki at http://wiki.alfresco.com.

Benchmark Results: Scalability of Alfresco Content Management System in a Unisys ES700/one Enterprise Server Environment, Unisys, December 2007.

McNabb, Kyle, *The Forrester Wave™: Enterprise Content Management Suites, Q4 2007*, Forrester Research, Inc., November 9, 2007.

McNabb, Kyle and Powers, Stephen, *The Forrester Wave™: Web Content Management For External Sites, Q3 2007*, Forrester Research, Inc., July 16, 2007.